



# Mech-DLK SDK用户手册

v2.0.0

# 目录

---

1. 欢迎 .....	1
2. 安装指南 .....	2
3. 快速入门 .....	3
4. 例程实操指南 .....	5
4.1. C 语言 (Windows) .....	5
5. API 参考手册 .....	11
6. 常见问题 .....	12

# 1. 欢迎

## 概述

Mech-DLK SDK 是专门配合 Mech-DLK 使用的二次开发软件包，主要用来帮助用户在已有的软件体系内轻松构建深度学习推理部分。用户可以在不依赖 Mech-Vision 的情况下，快速部署深度学习模型，灵活地集成深度学习功能到自己的应用中。目前支持 C 语言开发。

你可以应用 Mech-DLK SDK 实现 Mech-DLK 2.4.2 及以上版本导出模型的推理。

## 目录

本手册由以下几个部分构成，可根据需求查阅：

序号	章节	内容
1	<a href="#">安装指南</a>	查看系统要求及需要配置的环境、获取 Mech-DLK SDK。
2	<a href="#">快速入门</a>	了解如何使用 Mech-DLK SDK 进行缺陷分割模型的推理。
3	<a href="#">例程实操指南</a>	了解 C 语言例程类型及运行前提，构建及运行例程。
4	<a href="#">API 参考手册</a>	查看 C 语言 API 和结构体。
5	<a href="#">常见问题</a>	查看常见问题。

## 2. 安装指南

### 系统要求

建议应用 Mech-DLK SDK 进行模型推理的设备满足以下软硬件要求。

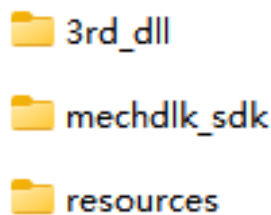
加密狗授权版本	Pro-Run	Pro-Train
操作系统	Windows 10 及以上	
CPU	Intel® Core™ i7-6700 及以上	
内存	8 GB 及以上	16 GB 及以上
显卡	GeForce GTX 1650 及以上	GeForce RTX 3060 及以上
显卡驱动	驱动版本 472.50 及以上	



Pro-Run 版本具有 Mech-DLK SDK 部署、标注、运行模式功能；Pro-Train 版本具有级联、标注、训练、验证以及 Mech-DLK SDK 部署功能。

### 获取 Mech-DLK SDK

1. 在本地创建一个项目文件夹，例如“dlk\_sdk”。
2. 将 [Mech-DLK SDK](#) 从 GitHub 克隆到该项目文件夹下。
3. 从[下载中心](#)下载 Mech-DLK SDK 依赖的第三方库（3rd\_dll.zip）和资源文件（resources.zip）到该项目文件夹下。
4. 解压第三方库压缩包和资源文件压缩包。此时，该项目文件夹应包含以下内容：



请勿随意更改文件夹里的文件，并记住文件的位置，供后续使用。

### 配置环境

如需快速运行 Mech-DLK SDK 中的 C 语言例程，请确保你的电脑上具备所需配置环境。你可以参考以下几点要求进行确认：

1. 安装了 [CMake](#) 和 [Visual Studio](#)（必需）
2. 添加了相关的环境变量

## 3. 快速入门

本章介绍如何应用 Mech-DLK SDK 实现 Mech-DLK 导出的缺陷分割模型的推理。

### 准备工作

- [安装Mech-DLK SDK](#)。
- 将梅卡曼德的加密狗插入电脑。

### 推理流程



### 函数说明

下面以 Mech-DLK 导出的缺陷分割模型为例，介绍应用 Mech-DLK SDK 实现模型推理时使用的函数。

#### 创建输入图像

```
MImage input;  
createImage("path/to/image.png", &input);
```

调用 createImage 函数创建输入图像。

#### 创建推理引擎

```
Engine engine;  
createPackInferEngine(&engine, "path/to/xxx.dlcpack", GpuDefault, 0);
```

调用 createPackInferEngine 函数创建推理引擎。



- 当部署电脑上配置有英伟达的独立显卡时，你可以将推理后端（即此函数中的第三个参数）设置为 GpuDefault 或 GpuOptimization。
  - 当此参数设置为 GpuOptimization 时，推理速度更快，但需要等待 1~5 min 的优化时间。
- 当部署电脑上未配置英伟达的独立显卡时，推理后端只能设置为 CPU。
- 此函数中，第四个参数为英伟达显卡 ID，如果只有一张显卡，填写 0 即可。当推理后端设置为 CPU 时，此参数无效。

## 深度学习引擎推理

```
infer(&engine, &input, 1);
```

调用 infer 函数进行深度学习引擎推理。



此函数中，参数 1 表示推理的图片数量，必须与 input 中的图片数量相同。

## 获取缺陷分割结果

```
DefectAndEdgeResult* defectAndEdgeResult = NULL;  
unsigned int resultNum = 0;  
getDefectSegmentationResult(&engine, 0, &defectAndEdgeResult, &resultNum);
```

调用 getDefectSegmentationResult 函数得到缺陷分割模型的结果。



此函数中，第二个参数 0 表示深度学习模型推理包中的模型索引。

- 如果是单模型推理包，该参数只能设置为 0；
- 如果为级联模型推理包，该参数需要根据模型推理包中算法的顺序进行设置。

## 结果可视化

```
resultVisualization(&engine, &input, 1);  
showImage(&input, "result");
```

调用 resultVisualization 函数将模型推理结果绘制到 input 中。



此函数中，参数 1 表示推理的图片数量，必须与 input 中的图片数量相同。

## 释放内存

```
releaseDefectSegmentationResult(&defectAndEdgeResult, resultNum);  
releaseImage(&input);  
releasePackInferEngine(&engine);
```

分别将模型推理结果、输入图片、引擎模型所占内存释放掉，防止内存泄露。

## 4. 例程实操指南

---

### C 语言 (Windows)

#### 4.1. C 语言 (Windows)

本章介绍如何在 Windows 系统上用 CMake 和 Visual Studio 配置 Mech-DLK SDK 中的 C 语言 API 的例程。

##### 例程简介

例程分为 2 类：**Basic** 和 **Advanced**。

- **Basic** 例程：使用 Mech-DLK 导出的各类单级模型推理单张图片、获取并可视化结果；
- **Advanced** 例程：级联模型推理、多图同时推理。

##### Basic

- [Classification](#)：图像分类模型推理示例；
- [DefectSegmentation](#)：缺陷分割模型推理示例；
- [FastPositioning](#)：快速定位模型推理示例；
- [InstanceSegmentation](#)：实例分割模型推理示例；
- [ObjectDetection](#)：目标检测模型推理示例。

##### Advanced

- [CascadeModel](#)：级联模型推理示例；
- [FolderImagesInfer](#)：依次推理文件夹中图像示例；
- [MultiImageInfer](#)：多图同时推理示例。

##### 使用前提

使用 Mech-DLK SDK 中的 C 语言例程，需先满足以下使用前提：

- 安装必需软件。
- 添加相关的环境变量。

##### 安装必需软件

使用 Mech-DLK SDK 的 C 语言例程，必须安装 Mech-DLK SDK、CMake 及 Visual Studio。

## 安装最新版本的 Mech-DLK SDK

请根据 [安装指南](#) 安装 Mech-DLK SDK 及其依赖的第三方库和资源文件。

### 安装 CMake (3.2 或以上版本)

1. 下载 CMake: 选择 Windows x64 Installer 右侧的安装包。

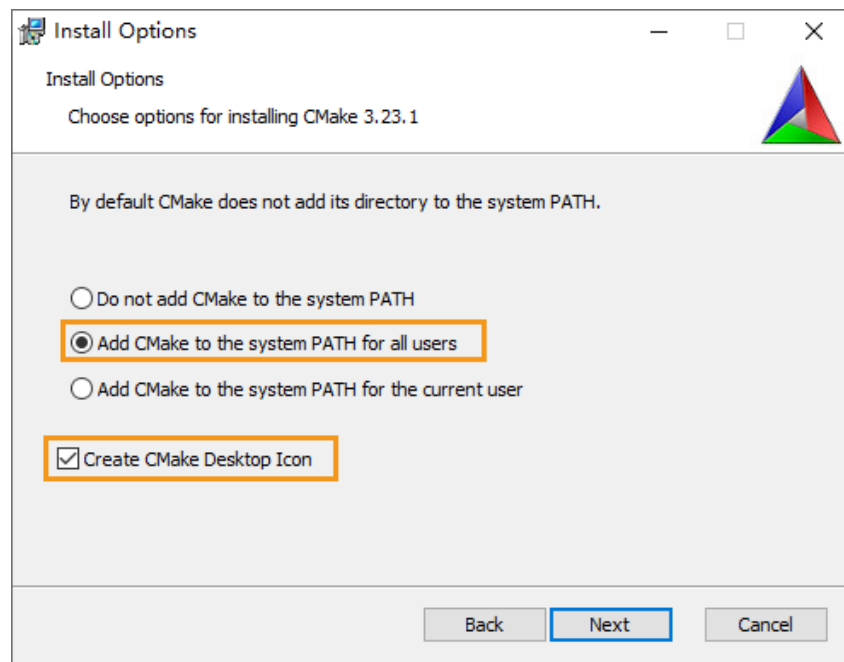
Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.23.1.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.23.1.zip

Binary distributions:

Platform	Files
Windows x64 Installer	cmake-3.23.1-windows-x86_64.msi
Windows x64 ZIP	cmake-3.23.1-windows-x86_64.zip
Windows I386 Installer	cmake-3.23.1-windows-i386.msi
Windows I386 ZIP	cmake-3.23.1-windows-i386.zip

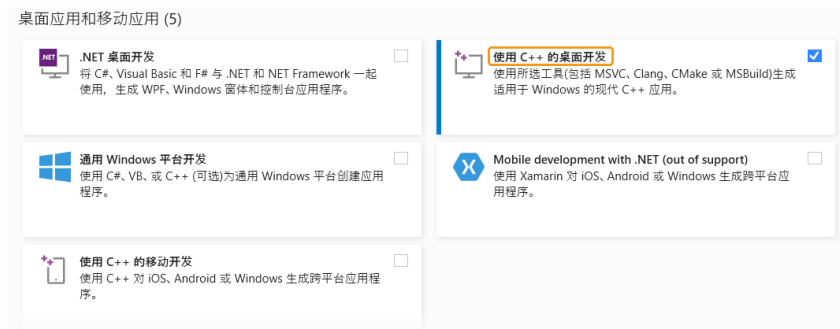
2. 安装时, 请勾选下图中的两个选项, 以将 CMake 添加至环境变量, 并创建 CMake 的桌面快捷方式。



### 安装 Visual Studio (2017 或以上版本)

1. 下载 [Visual Studio 安装包](#)。
2. 安装时, 请勾选 **桌面应用和移动应用** 分类中的“使用 C++ 的桌面开发”工作负荷, 再单击右下角的 [ 安装 ]。





## 添加环境变量

请根据以下步骤添加相关的环境变量。

1. 右键单击桌面上的 **此电脑**，选择 **属性**。
2. 选择 **高级系统设置**，在弹出的 **系统属性** 对话框的**高级选项卡**上，单击 **[ 环境变量 ]**，进入 **环境变量** 界面。
3. 在**系统变量**框中，滚动到 **Path** 并双击它进入到 **编辑环境变量** 页面。
4. 单击右上角 **[ 新建 ]**，添加 Mech-DLK SDK 的路径 `xxx\mechdlk_sdk`，然后单击 **[ 上移 ]** 按钮移动该路径至顶部，再单击右下角 **[ 确定 ]**。



添加环境变量后，请重启电脑，使添加的环境变量生效。

## 构建及运行例程

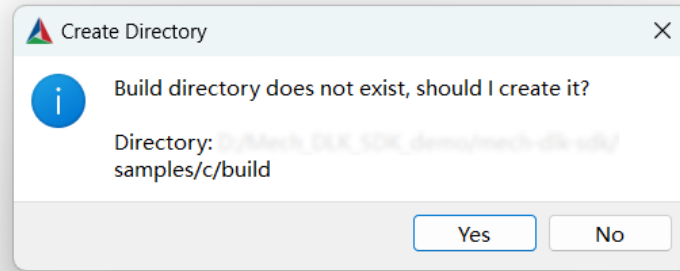
你可以一次性构建全部例程。

### 使用 CMake 配置例程

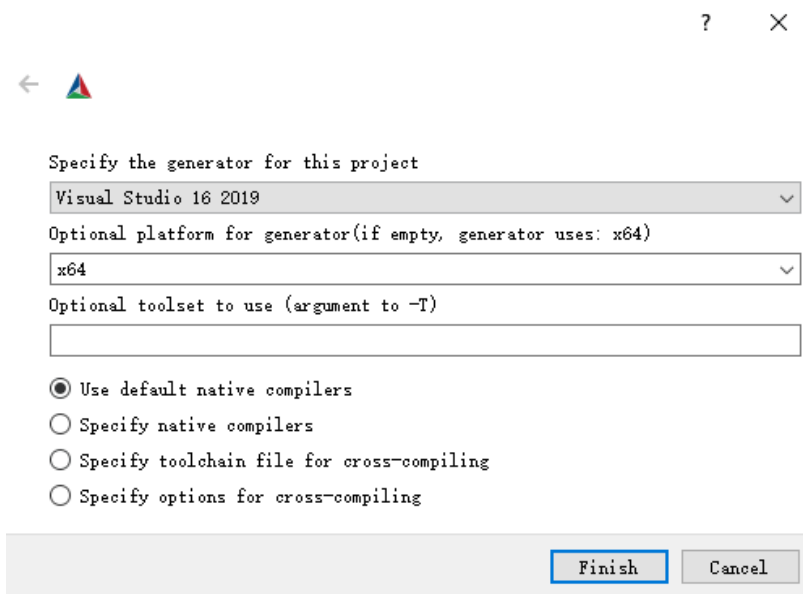
1. 双击 CMake 桌面图标打开软件。
2. 选择或输入源码路径与构建目录路径。如果没有构建目录路径，可以直接在源码路径后添加 `|build`，并继续以下的配置操作。

Where is the source code	<code>xxx\mechdlk_sdk\samples\c</code>
Where to build the binaries	<code>xxx\mechdlk_sdk\samples\c\build</code>

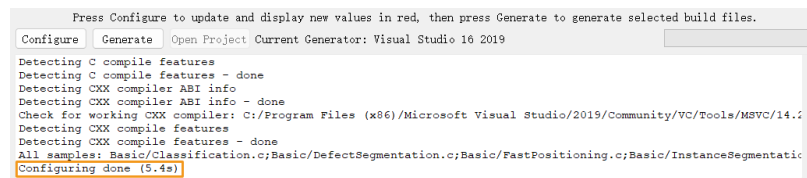
3. 单击 **[ Configure ]**。首次配置时，因没有构建目录路径，此时会弹窗请求构建 **build** 文件夹，请单击 **[ Yes ]**。



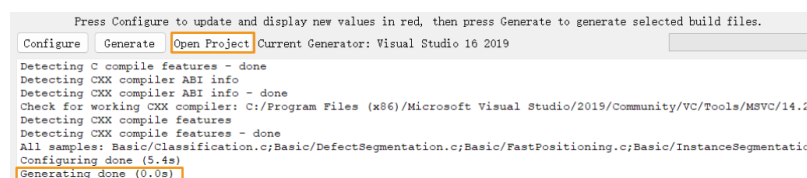
进入配置页面，选择 Visual Studio 的版本，然后单击 [ **Finish** ]。



配置成功后，日志最末行将显示 **Configuring done**。



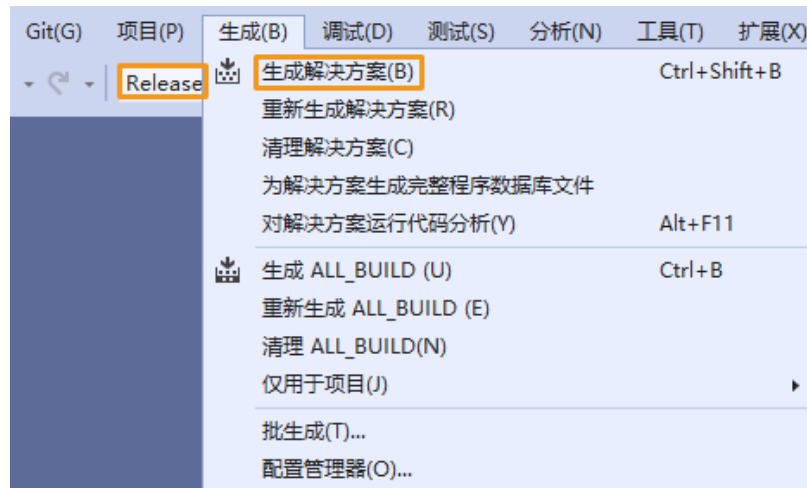
4. 单击 [ **Generate** ] 生成解决方案。生成成功后，日志最末行将显示 **Generating done**。然后，单击 [ **Open Project** ]，使用 Visual Studio 打开解决方案。



你也可以打开 **build** 文件夹（路径：`xxx\mechdlk_sdk\samples\c\build`），找到并双击 **MechDLKSDKCSamples.sln** 文件，使用 Visual Studio 打开解决方案。

## 使用 Visual Studio 构建例程

1. 确认 Visual Studio 工具栏中的解决方案配置为 **Release** 模式。当前未提供 **Debug** 版本的 DLL 文件。
2. 单击 生成 > 生成解决方案。每个例程会生成对应的 EXE 格式可执行文件，保存在 **Release** 文件夹中，位于 `xxx\mechdlk_sdk\samples\c\build` 目录下。



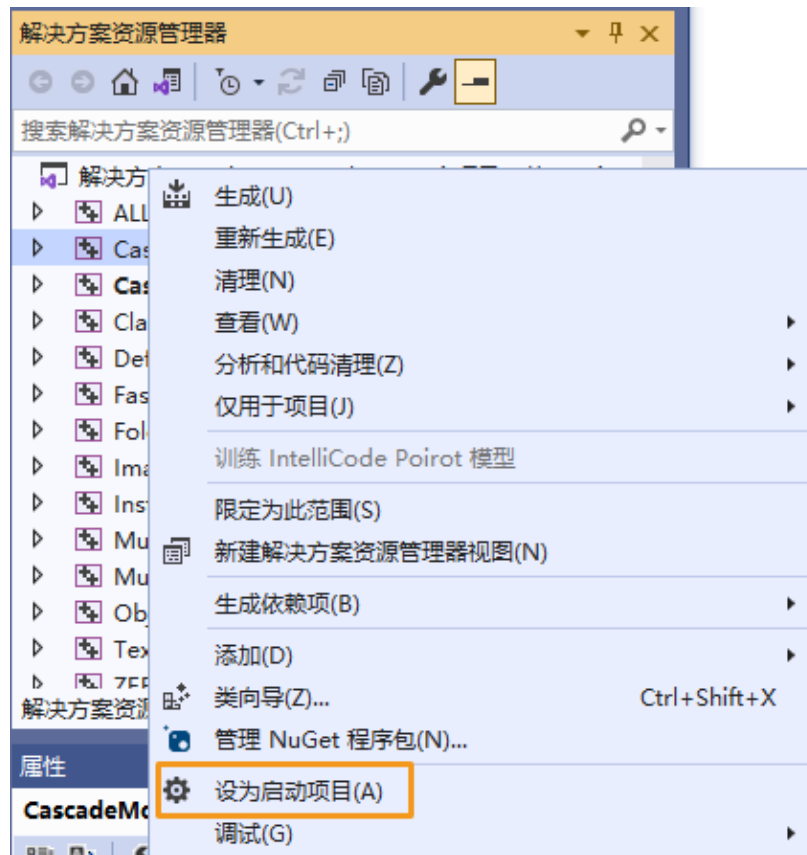
3. 将项目文件夹中的 **resources** 文件夹拷贝至以下目录下：
  - `xxx\mechdlk_sdk\samples\c\build\Release`
4. 将项目文件夹中的 **3rd\_dll** 文件夹内的全部文件、Mech-DLK SDK 文件夹 (`xxx/mechdlk_sdk/dll`) 路径下的全部文件拷贝至 `xxx\mechdlk_sdk\samples\c\build\Release` 目录下。

## 运行例程

你可以在 Visual Studio 中直接运行例程，也可以双击运行例程的可执行文件。

### 在 Visual Studio 中运行例程

1. 在 **解决方案资源管理器** 窗口中右键单击想要运行的例程，并选择 **设为启动项目**。



2. 单击工具栏中的  本地 Windows 调试器 运行该例程。

### 运行例程可执行文件

打开 **Release** 文件夹，双击运行与例程同名的 EXE 文件，即可得到运行结果。

## 5. API 参考手册

---

请单击 [Mech-DLK SDK C API](#) 查阅相关的 C 语言 API。

## 6. 常见问题

---

### Mech-DLK训练的模型可以在其他软件中使用吗？

Mech-DLK训练的模型可以在其他软件中使用，只需该软件满足如下条件：

- 该软件支持使用Mech-DLK SDK加载推理模型。
- 该软件能够调用相应语言API编写的程序来使用该模型。

### Mech-Vision/Mech-DLK SDK可以支持什么模型？

Mech-Vision/Mech-DLK SDK只能使用Mech-DLK训练出的模型或由梅卡曼德提供的超级模型，无法使用其他模型。

### Mech-DLK训练的模型能不能转换成其他格式的模型？

不能，模型格式目前不能进行转换。

### Mech-DLK SDK支持与哪些第三方软件集成？

Mech-DLK SDK目前支持与LABVIEW进行集成，通过加载C DLL的方式实现调用。除此之外，只要第三方软件能够支持调用使用C、C#或C++ API编写的程序块，就可以集成或调用Mech-DLK SDK。

### Mech-DLK SDK支持哪些语言的API？

目前支持C API。C#、C++以及Python API正在开发中，如需这些语言的API，请联系梅卡曼德技术支持。